1. 基础部分

1．编程求出1-100之间偶数的和。

public class Exam1{

public static void main(String args[]){

(1) ;//定义整型变量sum

for(int i=2;i<=100;){

sum+=i;

(2) ; }

System.out.println("1-100之间偶数的和是:"+sum);

}}

答案：（1）int sum =0;

（2）i=i+2

2．完成求n!的程序

public class Exam2{

public static void main(String args[]) {

factorial(3);

factorial(6);

factorial(9); }

void factorial(int n)  {

long m=1;

for (int x=1;x<=n;  (1)          )

  (2)              ;

System.out.println(n+"!="+m);}

}

答案：（1）x++

（2）m=m\*x

**3.** 以下程序要求从键盘输入一整数，判别该整数是否是素数，并输出“是素数”或“不是素数”，请将程序填写完整。

import java.util.\*;

public class PrimeExam {

public static void main(String[] args) throws IOException {

public static void main(String[] args){

Scanner sr= new Scanner( (1) );

int flag=0;

int x = sr. (2) ;

int y = (int)Math.sqrt(x);

for(int i=2;i<=y;i++){

if( (3) ){

System.out.println("不是素数");

flag=1;

break;

}

}

if( (4) )

System.out.println("是素数");

}

}

答案：（1）System.in

（2）nextInt()

1. x % i ==0

(4) flag==0

4、以下程序要求从键盘输入一个整数，判别该整数为几位数，并输出结果，请将下面的程序填写完整。

**public** **class** Blank1 {

**public** **static** **void** main(String[] args) **throws** IOException {

Scanner sc=**new** Scanner( （1） );

**int** count=0,t;

**int** x=sc.nextInt();

sc.close();

t=x;

**while**(t!=0){

count++;

（2） ;

}

System.***out***.println(x+"是"+count+"位数。");

}

}

答案：（1）System.***in***

（2）t=t/10

5.下面的程序通过方法调用从包含7个学号的数组中随机抽取一个学号并输出显示，请将下面的程序填写完整。

public class Ex1 {

public （1） String getXh(){

String[] xhs = {"201001","201002","201003","201004","201005","201006","201007"};

int index = （2） ; //生成0~6之间的随机数。

return xhs[index];

}

public static void main(String[] args){

System.out.println("随机抽取的学号为："+ （3） );

}

}

答案：（11）**static**

（12）(int)(Math.random()\*7)

（13）getXh()

**6、**根据程序注释提示将下面的程序填写完整。

class StringExample

{ public static void main(String args[])

{ String s1=new String("2012");

String s2=new String("100.50");

int x = （1） ; // 将s1转换为int类型

double y **=**  （2） ; // 将s2转换为double类型

double z = x + y;

String s3 = （3） ; //将z转换为字符串

StringBuffer sbr = new StringBuffer("Thingking");

String s4 = new String("in Java");

（4） ; //将s4连接在sbr的后面

System.out.println(sbr.toString()); //显示为 Thingking in Java

}

}

答案：（1）Integer.parseInt(s1)

（2）Double.parseDouble(s2)

（3）new String(z); 或 String.valueOf(z); 或 z+””

（4）sbr.append(s4)

二、第6、7、8章复习

7、在下面的程序中使用方法重载分别实现了两个和三个整数的相加，请将下面的程序填写完整。

复习：1）方法重载的概念 p98-99

2）方法重载的编写 p98-99 例题7.3

3) 默认构造方的概念p101

4) 调用默认构造方法创建对象p101，p85例题6.1

**class** AddOver{

**public （1）** {

**return** a+b;

}

**public** **int** add( **int** a, **int** b, **int** c ){

**return** a+b+c ;

}

}

**public** **class** Blank2{

**public** **static** **void** main(String[] args){

AddOver a= （2） ;

System.***out***.println(a.add(1,2));

System.***out***.println(a.add(1,2,3));

}

}

答案：（1）**int** add(**int** a,**int** b)

（2）**new** AddOver()

8.

复习1）构造方法的重载 p101-p104例题7.5

2） 从一个构造方法调用另一个构造方法 p101-p104例题7.6

class Apple

{

private String color;

private int num;

public Apple(String c, int n) {

color = c ;

num = n;

}

public Apple( String c) {

（1） ;//调用两个参数的构造方法，num默认为0

}

public Apple( ) {

（2） ; //调用一个参数的构造方法，color默认为"Unknown"

}

public String getColor() {

return color;

}

public int getNum() {

return num;

}

**}**

答案：（1）this(c,0)

（2）this("Unknown")

9、 构造一个类来描述一个点，该类的构成包括点的x和y两个坐标，以及一些对点进行的操作，包括：取得点的坐标值，利用另一个点对当前点的坐标进行赋值，请将下面的程序填写完整。

复习：1）构造方法的概念 p99-101例题7.4

2） 引用变量作为方法的参数、作为方法的返回值 p112-113 例题7.11、例题7.12、

class Point{

int x,y;

public （1） (int x, int y){

this.x=x;

this.y=y;

}

public Point getPoint(){

Point temp = new Point(0,0);

temp.x=x;

temp.y=y;

return （2） ;

}

public void setPoint( （3） ){

this.x=s.x;

this.y=s.y;

}

}

public class Blank3 {

public static void main(String[] args){

Point a = new Point(3,4);

Point b = new Point(0,0);

b=a.getPoint();

Point c = new Point(0,0);

c.setPoint(b);

}

}

答案： （1）Point

（2）temp

（3）Point s

10、向方法内传递对象

【例7.11】以圆柱体类Cylinder的对象为参数进行方法调用，并说明对象的比较。

复习：1）构造方法的概念 p99-101例题7.4

2） 引用变量作为方法的参数、作为方法的返回值 p112-113 例题7.11、例题7.12、

//filename：App7\_11.java

class Cylinder //定义类Cylinder

{

private static double pi=3.14;

private double radius;

private int height;

（1） //设计构造方法，有两个参数

{

radius=r;

height=h;

}

public void compare( （2） ) //以对象作为方法的参数

{

if (this==v) //判断this与v是否指向同一对象

System.out.println("这两个对象相等");

else

System.out.println("这两个对象不相等");

}

}

public class App7\_11 //主类

{

public static void main(String[] args)

{

Cylinder volu1=new Cylinder(1.0,2);

Cylinder volu2=new Cylinder(1.0,2);

Cylinder volu3= volu1;

volu1.compare(volu2); //调用compare()，比较volu1与volu2是否相等

volu1.compare(volu3); //调用compare()，比较volu1与volu3是否相等

}

}

答案：（1）public Cylinder(double r, int h)

（2）Cylinder volu

**11** *方法的返回值为对象*

【例7.12】创建人类Person，在该类中定义一个以对象作为返回值的方法compare()。

复习：1）构造方法的概念 p99-101例题7.4

2） 引用变量作为方法的参数、作为方法的返回值 p112-113 例题7.11、例题7.12、

//filename：App7\_12.java

class Person //定义类Person

{

private String name;

private int age;

（1） //设计构造方法，有两个参数

{

this.name=name;

this.age=age;

}

public Person compare( （2） ) //以对象作为方法的参数

{

if (this.age>p.age)

return this; //返回调用该方法的对象

else

return （3） ; //返回参数对象

}

}

public class App7\_12 //主类

{

public static void main(String[] args)

{

Person per1=new Person ("张三",20);

Person per2=new Person ("李四",21);

Person per3;

per3= per1.compare(per2);

if (per3==per1)

System.out.println("张三年龄大");

else

System.out.println("李四年龄大");

}

}

答案：（1）public Person(String name, int age)

（2） Person p

（3）p

12、【例6.5】以一维数组为参数的方法调用，求若干数的最小值。

复习 2） 数组作为方法的参数 p91 例题6.5

//filename：App6\_5.java 以数组为参数的方法调用

public class App6\_5 //定义主类

{

public static void main(String[] args)

{

int[] a={8,3,7,88,9,23}; //定义一维数组a

LeastNumb minNumber=new LeastNumb ();

minNumber.least( （1） ); //将一维数组a传入least()方法

}

}

class LeastNumb //定义另一个类

{

public void least(int[] array) //参数array接收一维整型数组

{

int temp=array[0];

for (int i=1;i< （2） ;i++)

if (temp>array[i])

temp=array[i];

System.out.println("最小的数为："+temp);

}

}

答案：（1）a

（2）array.length

13 【例8.11】

复习：1）接口的概念 p134-137例题8.11

2）抽象方法

3）常量定义

4）定义一个类实现一个接口

5）方法覆盖（方法重写）

//filename：App8\_11.java 接口的实现

（1） //定义接口Shape

{

（2） //声明常量PI，值为314

（3） //声明抽象方法getArea

}

（4） //定义Circle类实现接口Shape

{

double radius;

public Circle(double r)

{

radius=r;

}

（5） //实现接口中的getArea()方法

{

return pi\*radius\*radius;

}

}

public class App8\_11 //主类

{

public static void main(String[] args)

{

Shape circle=new Circle(5.0); //声明父接口变量circle，指向子类对象

System.out.print("圆面积="+circle.getArea());

}

}

答案：

（1）interface Shape

（2）public static final double PI=3.14;

（3）public abstract double getArea();

（4）class Circle implements Shape

（5）public double getArea()

三、第9、10章复习 异常处理与输入输出

14. 下面的程序完成从D:\Hello.txt中读取文本并显示在屏幕上，请将下面的程序填写完整。

复习p186-p192 10.3 Reader和Writer流类例题10.5、例题10.6、例题10.7、例题10.8

import java.io.\*;

**class** Blank4 {

**public** **static** **void** main ( String[] args ) {

String fileName = "D:\\Hello.txt", line;

**try** {

BufferedReader in = **new** BufferedReader( （1） );

line = in.readLine();

**while** ( （2） ) {

System.***out***.println( line );

line = （3） ;

}

（4） //关闭文件

}

**catch** (IOException e) {

System.***out***.println("Problem reading " + fileName );

}

}

}

答案：（1）**new** FileReader(fileName)

（2）line!=**null**

（3）in.readLine()

（4）in.close();

15. 下面的程序利用缓冲区输入输出流，完成文件的复制（拷贝到D盘根目录上） ，请将下面的程序填写完整。

复习p186-p192 10.3 Reader和Writer流类例题10.5、例题10.6、例题10.7、例题10.8

//filename：App10\_8.java

**import java.io.\*;**

**public class App10\_8**{

**public static void main(String[] args) throws IOException** {

String str=**new String();**

**BufferedReader in = null;**

**BufferedWriter out = null;**

**try**

{

in=**new BufferedReader(new FileReader("d:\\java\\test.txt"));**

out=**new BufferedWriter(** （1） **);**

**while ((str=in.readLine())!=null)**

{

System.***out.println(str);***

（2） ; //将读取到的一行数据写入到缓冲区输出流对象中

out.newLine();//写入回车换行符

}

（3） ; //将缓冲区中的数据全部写入到文件中

in.close();

out.close();

}

**catch (IOException ioe)** {

System.***out.println("错误! "+ioe);*** }

}

}

答案：（1）**new FileWriter("d:\\test.txt")**

（2）out.write(str);

（3）out.flush();

**16.** 下面的程序使用BufferedWriter类在D:\Hello.txt文件中写入10万个数并输出所用的时间，请将程序填写完整。

复习p186-p192 10.3 Reader和Writer流类例题10.5、例题10.6、例题10.7、例题10.8

import java.io.\*;

public class Ex4 {

public static void main(String[] args) throws IOException{

long t = System.currentTimeMillis();

BufferedWriter fw =new BufferedWriter( （1） );

for (int i=1;i<=100000;i++){

（2） ( i +" \n");

}

fw.close();

t=System.currentTimeMillis()-t;

System.out.println("Time elapsed: "+t +"ms");

}

}

答案：（1）new FileWriter(“D:\\Hello.txt”)

（2）fw.write